MAB Learning in IoT Networks

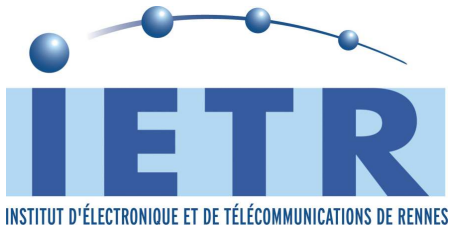
Decentralized Multi-Player Multi-Arm Bandits

Lilian Besson

Advised by Christophe Moy Émilie Kaufmann

PhD Student
Team SCEE, IETR, CentraleSupélec, Rennes
& Team SequeL, CRIStAL, Inria, Lille

SCEE Seminar - 23 November 2017



Motivation: *Internet of Things* problem

A *lot* of IoT devices want to access to a single base station.

- Insert them in a possibly **crowded wireless network**.
- With a protocol **slotted in both time and frequency**.
- Each device has a **low duty cycle** (a few messages per day).

Motivation: *Internet of Things* problem

A *lot* of IoT devices want to access to a single base station.

- Insert them in a possibly **crowded wireless network**.
- With a protocol **slotted in both time and frequency**.
- Each device has a **low duty cycle** (a few messages per day).

Goal

- Maintain a **good Quality of Service**.
- **Without** centralized supervision!

Motivation: *Internet of Things* problem

A *lot* of IoT devices want to access to a single base station.

- Insert them in a possibly **crowded wireless network**.
- With a protocol **slotted in both time and frequency**.
- Each device has a **low duty cycle** (a few messages per day).

Goal

- Maintain a **good Quality of Service**.
- **Without** centralized supervision!

How?

- Use **learning algorithms**: devices will learn on which frequency they should talk!

Outline and references

1. Introduction and motivation
2. Model and hypotheses
3. Baseline algorithms : to compare against naive and efficient centralized approaches
4. Two Multi-Armed Bandit algorithms : UCB, TS
5. Experimental results
6. An easier model with theoretical results
7. Perspectives and future works

Main references are my recent articles (on HAL):

- *Multi-Armed Bandit Learning in IoT Networks and non-stationary settings*, Bonnefoi, Besson, Moy, Kaufmann, Palicot. CrownCom 2017,
- *Multi-Player Bandits Models Revisited*, Besson, Kaufmann. arXiv:1711.02317,

First model

- Discrete time $t \geq 1$ and K radio channels (*e.g.*, 10) (*known*)

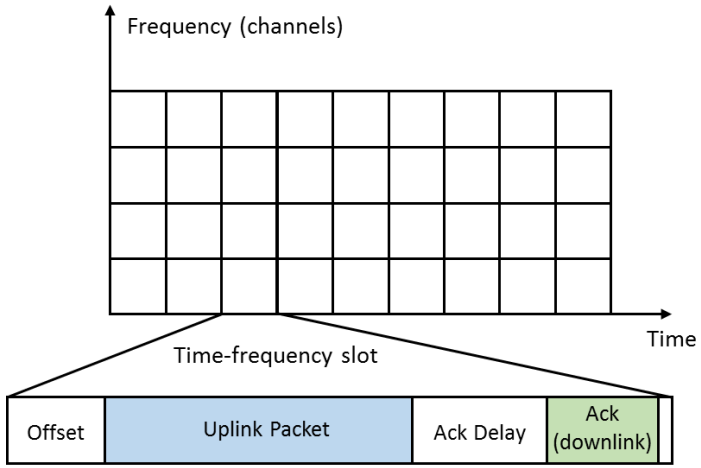


Figure 1: Protocol in time and frequency, with an *Acknowledgement*.

- D **dynamic** devices try to access the network *independently*
- $S = S_1 + \cdots + S_K$ **static** devices occupy the network : $S_1, \ldots, S_K$ in each channel (*unknown*)

Hypotheses I

Emission model

- Each device has the same *low* emission probability: each step, each device sends a packet with probability p. (this gives a duty cycle proportional to $1/p$)

Background traffic

- Each static device uses only one channel.
- Their repartition is fixed in time.

$\Longrightarrow$ *Background traffic, bothering the dynamic devices!*

Hypotheses II

Dynamic radio reconfiguration

- Each **dynamic device decides the channel it uses to send every packet**.
- It has memory and computational capacity to implement simple **decision algorithm**.

Problem

- *Goal* : *minimize packet loss ratio* (= maximize number of received `Ack`) in a *finite-space discrete-time Decision Making Problem*.
- *Solution ?* **Multi-Armed Bandit algorithms, decentralized** and used **independently** by each device.

A naive strategy : uniformly random access

- **Uniformly random access**: dynamic devices choose uniformly their channel in the pull of K channels.
- Natural strategy, dead simple to implement.
- Simple analysis, in term of **successful transmission probability** (for every message from dynamic devices) :

$$\mathbb{P}(\text{success}|\text{sent}) = \sum_{i=1}^{K} \underbrace{(1-p/K)^{D-1}}_{\text{No other dynamic device}} \times \underbrace{(1-p)^{S_i}}_{\text{No static device}} \times \frac{1}{K}.$$

A naive strategy : uniformly random access

- **Uniformly random access**: dynamic devices choose uniformly their channel in the pull of K channels.
- Natural strategy, dead simple to implement.
- Simple analysis, in term of **successful transmission probability** (for every message from dynamic devices) :

$$\mathbb{P}(\text{success}|\text{sent}) = \sum_{i=1}^{K} \underbrace{(1-p/K)^{D-1}}_{\text{No other dynamic device}} \times \underbrace{(1-p)^{S_i}}_{\text{No static device}} \times \frac{1}{K}.$$

No learning

- Works fine only if all channels are similarly occupied, but **it cannot learn** to exploit the best (more free) channels.

Optimal centralized strategy I

- If an oracle can decide to affect D_i dynamic devices to channel i, the **successful transmission probability** is:

$$\mathbb{P}(\text{success}|\text{sent}) = \sum_{i=1}^{K} \underbrace{(1-p)^{D_i-1}}_{D_i-1 \text{ others}} \times \underbrace{(1-p)^{S_i}}_{\text{No static device}} \times \underbrace{D_i/D}_{\text{Sent in channel } i}.$$

- The oracle has to solve this **optimization problem**:

$$\begin{cases} \arg\max_{D_1,\dots,D_K} & \sum_{i=1}^{K} D_i (1-p)^{S_i+D_i-1} \\ \text{such that} & \sum_{i=1}^{K} D_i = D \text{ and } D_i \geq 0, \quad \forall 1 \leq i \leq K. \end{cases}$$

- We solved this quasi-convex optimization problem with *Lagrange multipliers*, only numerically.

Optimal centralized strategy II

- $\Longrightarrow$ Very good performance, maximizing the transmission rate of all the D dynamic devices

But unrealistic

But **not achievable in practice**: no centralized control and no oracle!

Now let see *realistic decentralized approaches*

$\hookrightarrow$ Machine Learning ?

$\hookrightarrow$ Reinforcement Learning ?

$\hookrightarrow$ *Multi-Armed Bandit* !

Multi-Armed Bandit formulation

A dynamic device tries to collect *rewards* when transmitting :

- it transmits following a Bernoulli process (probability p of transmitting at each time step t),
- chooses a channel $A(\tau) \in \{1, \ldots, K\}$,
  - if `Ack` (no collision) $\implies$ reward $r_{A(\tau)} = 1$,
  - if collision (no `Ack`) $\implies$ reward $r_{A(\tau)} = 0$.

Reinforcement Learning interpretation

Maximize transmission rate $\equiv$ **maximize cumulated rewards**

$$\max_{\text{algorithm } A} \sum_{\tau=1}^{\text{horizon}} r_{A(\tau)}.$$

Upper Confidence Bound algorithm (UCB_1)

Dynamic device keep τ number of sent packets, $T_k(\tau)$ selections of channel k, $X_k(\tau)$ successful transmission in channel k.

1. For the first K steps ($\tau = 1, \ldots, K$), try each channel *once*.
2. Then for the next steps $t > K$:
   - Compute the index $g_k(\tau) := \underbrace{\frac{X_k(\tau)}{T_k(\tau)}}_{\text{Mean } \widehat{\mu_k}(\tau)} + \underbrace{\sqrt{\frac{\log(\tau)}{2T_k(\tau)}}}_{\text{Upper Confidence Bound}},$
   - Choose channel $A(\tau) = \arg\max_k g_k(\tau)$,
   - Update $T_k(\tau + 1)$ and $X_k(\tau + 1)$.

References: [Lai & Robbins, 1985], [Auer et al, 2002], [Bubeck & Cesa-Bianchi, 2012]

Thompson Sampling : Bayesian approach

A dynamic device assumes a stochastic hypothesis on the background traffic, modeled as Bernoulli distributions.

- Rewards $r_k(\tau)$ are assumed to be *i.i.d.* samples from a Bernoulli distribution $\mathrm{Bern}(\mu_k)$.
- A **binomial Bayesian posterior** is kept on the mean availability μ_k : $\mathrm{Bin}(1 + X_k(\tau), 1 + T_k(\tau) - X_k(\tau))$.
- Starts with a *uniform prior* : $\mathrm{Bin}(1, 1) \sim \mathcal{U}([0, 1])$.

1. Each step $\tau \geq 1$, draw a sample from each posterior $i_k(\tau) \sim \mathrm{Bin}(a_k(\tau), b_k(\tau))$,
2. Choose channel $A(\tau) = \arg\max_k \ i_k(\tau)$,
3. Update the posterior after receiving `Ack` or if collision.

References: [Thompson, 1933], [Kaufmann et al, 2012]

Experimental setting

Simulation parameters

- $K = 10$ channels,
- $S + D = 10000$ devices **in total**. Proportion of dynamic devices $D/(S + D)$ varies,
- $p = 10^{-3}$ probability of emission, for all devices,
- Horizon $= 10^6$ time slots, ($\simeq 1000$ messages / device)
- Various settings for $(S_1, \ldots, S_K)$ static devices repartition.

What do we show (for static S_i)

- After a short learning time, MAB algorithms are almost as efficient as the oracle solution !
- Never worse than the naive solution.
- Thompson sampling is more efficient than UCB.
- Stationary alg. outperform adversarial ones (UCB $\gg$ Exp3).

10% of dynamic devices

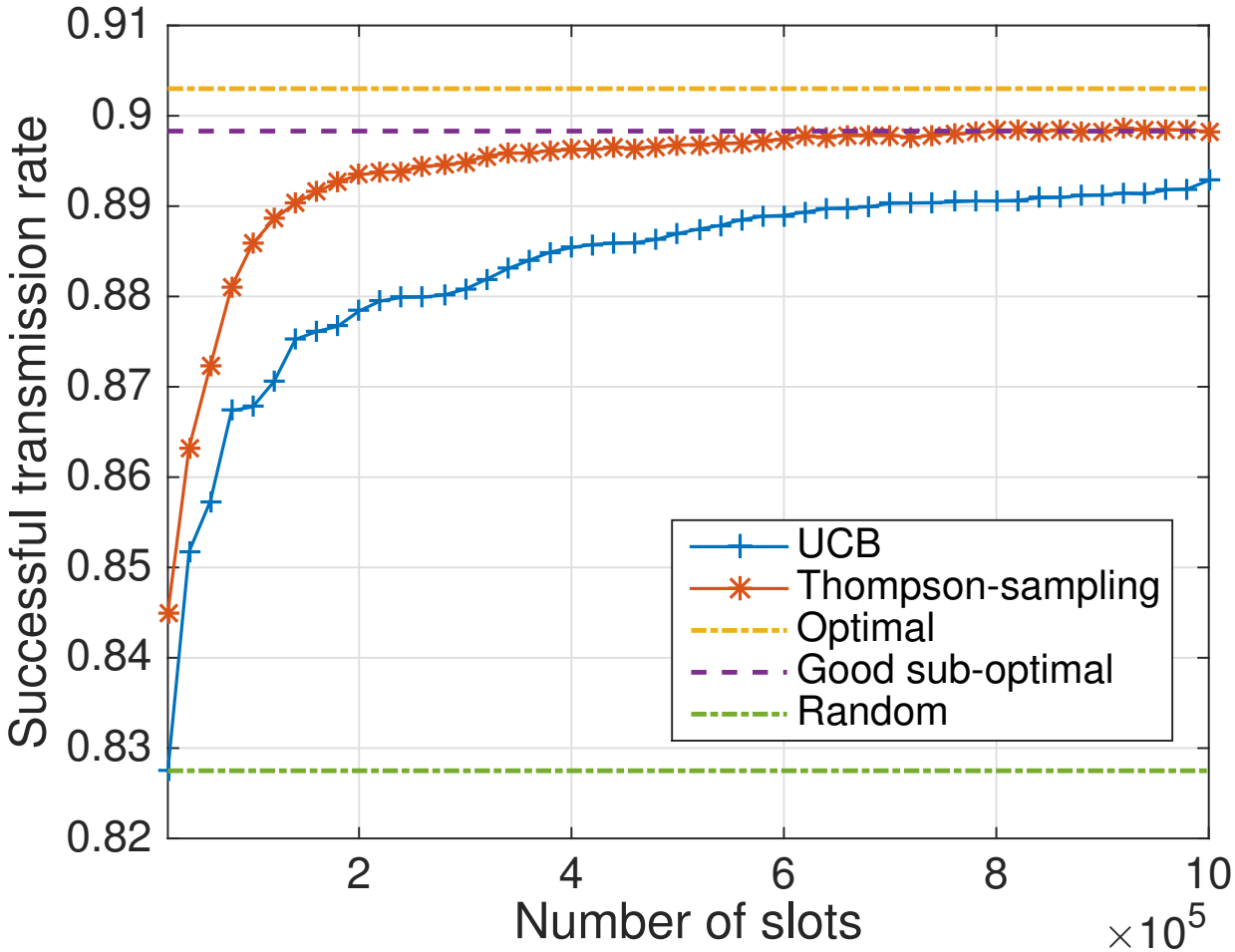


Figure 2: 10% of dynamic devices. 7% of gain.

30% of dynamic devices

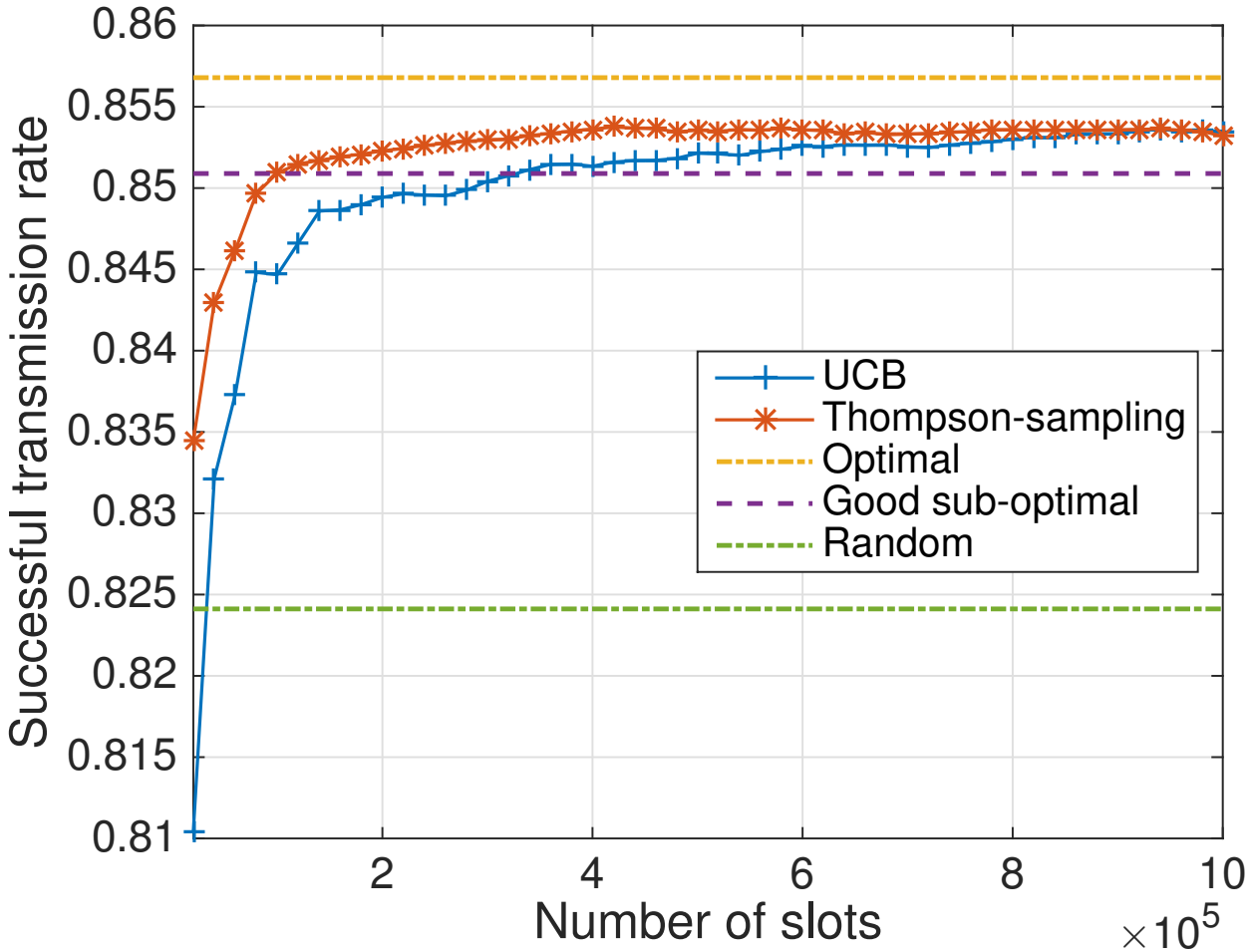


Figure 3: 30% of dynamic devices. 3% of gain but not much is possible.

Dependence on $D/(S+D)$

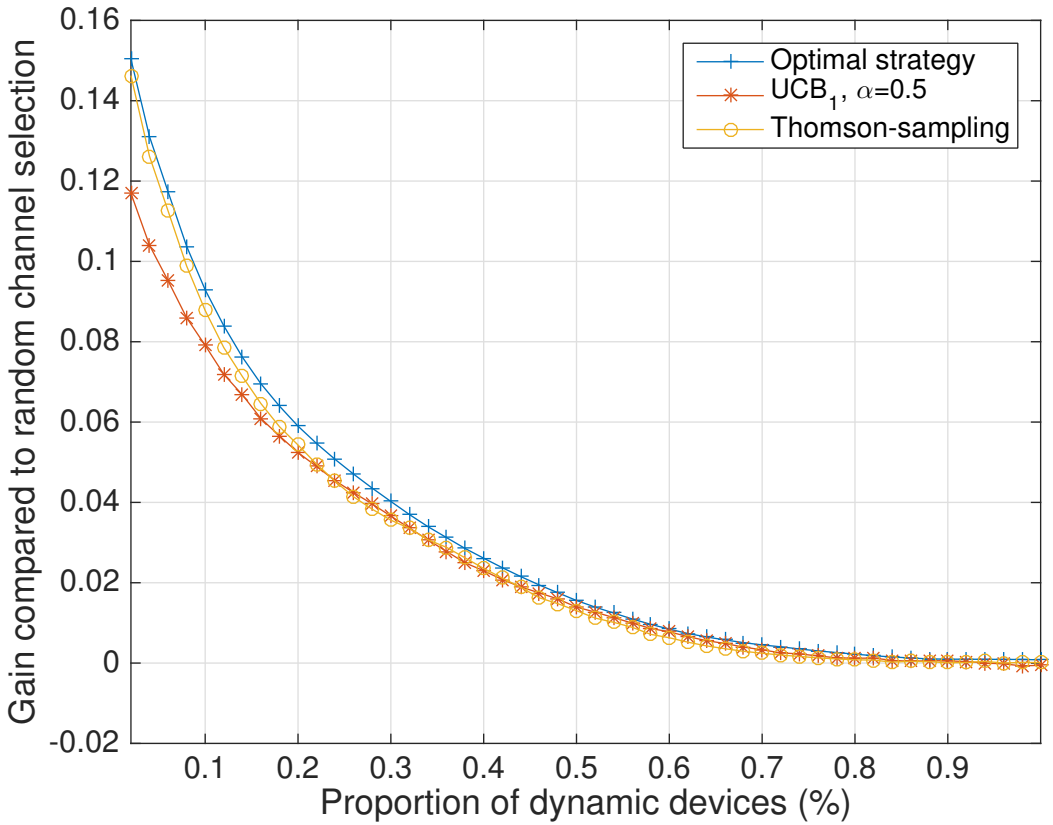


Figure 4: *Almost optimal*, for any proportion of dynamic devices, *after a short learning time*. Up-to 16% gain over the naive approach!

Section 6

A brief presentation of a different approach...

Theoretical results for an easier model

An easier model

Easy case

- $M \leq K$ dynamic devices **always communicating** ($p = 1$).
- Still interesting: many mathematical and experimental results!

An easier model

Easy case

- $M \leq K$ dynamic devices **always communicating** ($p = 1$).
- Still interesting: many mathematical and experimental results!

Two variants

- *With sensing*: Device first senses for presence of Primary Users (background traffic), then use `Ack` to detect collisions. Model the "classical" Opportunistic Spectrum Access problem. Not exactly suited for IoT networks like LoRa or SigFox, can model ZigBee, and can be analyzed mathematically...

  (*cf* Wassim's and Navik's theses, 2012, 2017)
- *Without sensing*: like our IoT model but smaller scale. Still very hard to analyze mathematically.

Notations for this second model

Notations

- K channels, modeled as Bernoulli $(0/1)$ distributions of mean $\mu_k =$ background traffic from *Primary Users*,
- M devices use channel $A^j(t) \in \{1, \dots, K\}$ at each time step,
- Reward: $r^j(t) := Y_{A^j(t),t} \times \mathbb{1}(\overline{C^j(t)}) = \mathbb{1}(\text{uplink \& } \texttt{Ack})$
  - with sensing information $Y_{k,t} \sim \mathrm{Bern}(\mu_k)$,
  - collision for device j $C^j(t) = \mathbb{1}(\textit{alone on arm } A^j(t))$.

Notations for this second model

Notations

- K channels, modeled as Bernoulli $(0/1)$ distributions of mean $\mu_k =$ background traffic from *Primary Users*,
- M devices use channel $A^j(t) \in \{1, \dots, K\}$ at each time step,
- Reward: $r^j(t) := Y_{A^j(t),t} \times \mathbb{1}(\overline{C^j(t)}) = \mathbb{1}(\text{uplink \& } \texttt{Ack})$
  - with sensing information $Y_{k,t} \sim \mathrm{Bern}(\mu_k)$,
  - collision for device j $C^j(t) = \mathbb{1}(\textit{alone on arm } A^j(t))$.

Goal : *decentralized* reinforcement learning optimization!

- Each player wants to **maximize its cumulated reward**,
- With no central control, and no exchange of information,
- Only possible if : each player converges to one of the M best arms, orthogonally (without collisions)

Centralized regret

New measure of success

- Not the network throughput or collision probability,
- Now we study the **centralized regret**

$$R_T(\boldsymbol{\mu}, M, \rho) := \left(\sum_{k=1}^{M} \mu_k^*\right) T - \mathbb{E}_\mu \left[\sum_{t=1}^{T} \sum_{j=1}^{M} r^j(t)\right].$$

Centralized regret

New measure of success

- Not the network throughput or collision probability,
- Now we study the **centralized regret**

$$R_T(\boldsymbol{\mu}, M, \rho) := \left(\sum_{k=1}^{M} \mu_k^*\right) T - \mathbb{E}_\mu \left[\sum_{t=1}^{T}\sum_{j=1}^{M} r^j(t)\right].$$

Two directions of analysis

- Clearly $R_T = \mathcal{O}(T)$, but we want a sub-linear regret
- *What is the best possible performance of a decentralized algorithm in this setting?*
  $\hookrightarrow$ **Lower Bound** on regret for **any** algorithm !
- *Is this algorithm efficient in this setting?*
  $\hookrightarrow$ **Upper Bound** on regret for **one** algorithm !

Asymptotic Lower Bound on regret I

For any algorithm, decentralized or not, we have

$$R_T(\boldsymbol{\mu}, M, \rho) = \sum_{k \in M\text{-worst}} (\mu^*_M - \mu_k)\mathbb{E}_\mu[T_k(T)]$$
$$+ \sum_{k \in M\text{-best}} (\mu_k - \mu^*_M)(T - \mathbb{E}_\mu[T_k(T)]) + \sum_{k=1}^{K} \mu_k \mathbb{E}_\mu[\mathcal{C}_k(T)].$$

Small regret can be attained if...

1. Devices can quickly identify the bad arms M-worst, and not play them too much (*number of sub-optimal selections*),
2. Devices can quickly identify the best arms, and most surely play them (*number of optimal non-selections*),
3. Devices can use orthogonal channels (*number of collisions*).

Asymptotic Lower Bound on regret II

Lower-bounds

- The first term $\mathbb{E}_\mu[T_k(T)]$, for sub-optimal arms selections, is lower-bounded, using technical information theory tools (Kullback-Leibler divergence, entropy),
- And we lower-bound collisions by... 0 : hard to do better!

Theorem 1 [Besson & Kaufmann, 2017]

- For any uniformly efficient decentralized policy, and any non-degenerated problem $\boldsymbol{\mu}$,

$$\liminf_{T\to+\infty} \frac{R_T(\boldsymbol{\mu}, M, \rho)}{\log(T)} \geq M \times \left(\sum_{k \in M\text{-worst}} \frac{(\mu^*_M - \mu_k)}{\mathrm{kl}(\mu_k, \mu^*_M)} \right).$$

Where $\mathrm{kl}(x, y) := x \log(\frac{x}{y}) + (1-x) \log(\frac{1-x}{1-y})$ is the binary Kullback-Leibler divergence.

Illustration of the Lower Bound on regret

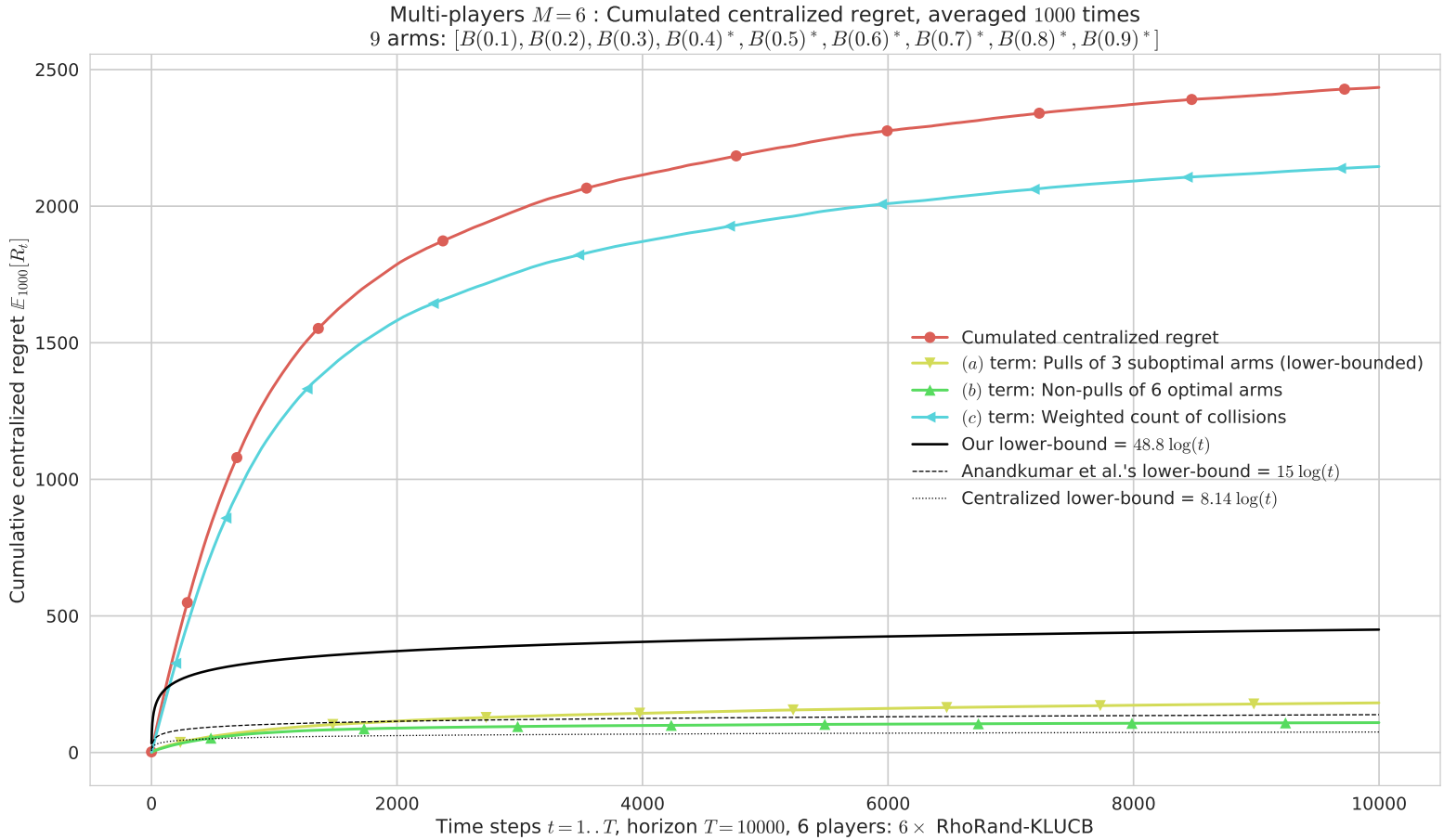


Figure 5: Any such lower-bound is very asymptotic, usually not satisfied for small horizons. We can see the importance of the collisions!

Algorithms for this easier model

Building blocks : separate the two aspects

1. **MAB policy** to learn the best arms (use sensing $Y_{A^j(t),t}$),
2. **Orthogonalization scheme** to avoid collisions (use $C^j(t)$).

Algorithms for this easier model

Building blocks : separate the two aspects

1. **MAB policy** to learn the best arms (use sensing $Y_{A^j(t),t}$),
2. **Orthogonalization scheme** to avoid collisions (use $C^j(t)$).

Many different proposals for *decentralized* learning policies

- Recent: MEGA and Musical Chair, [Avner & Mannor, 2015], [Shamir et al, 2016]
- State-of-the-art: **RhoRand policy** and variants, [Anandkumar et al, 2011]
- **Our proposals**: [Besson & Kaufmann, 2017]
  - With sensing: RandTopM and MCTopM are sort of mixes between RhoRand and Musical Chair, using UCB indexes or more efficient index policy (kl-UCB),
  - Without sensing: Selfish use a UCB index directly on the reward $r^j(t)$: like the first IoT model !

Illustration of different algorithms

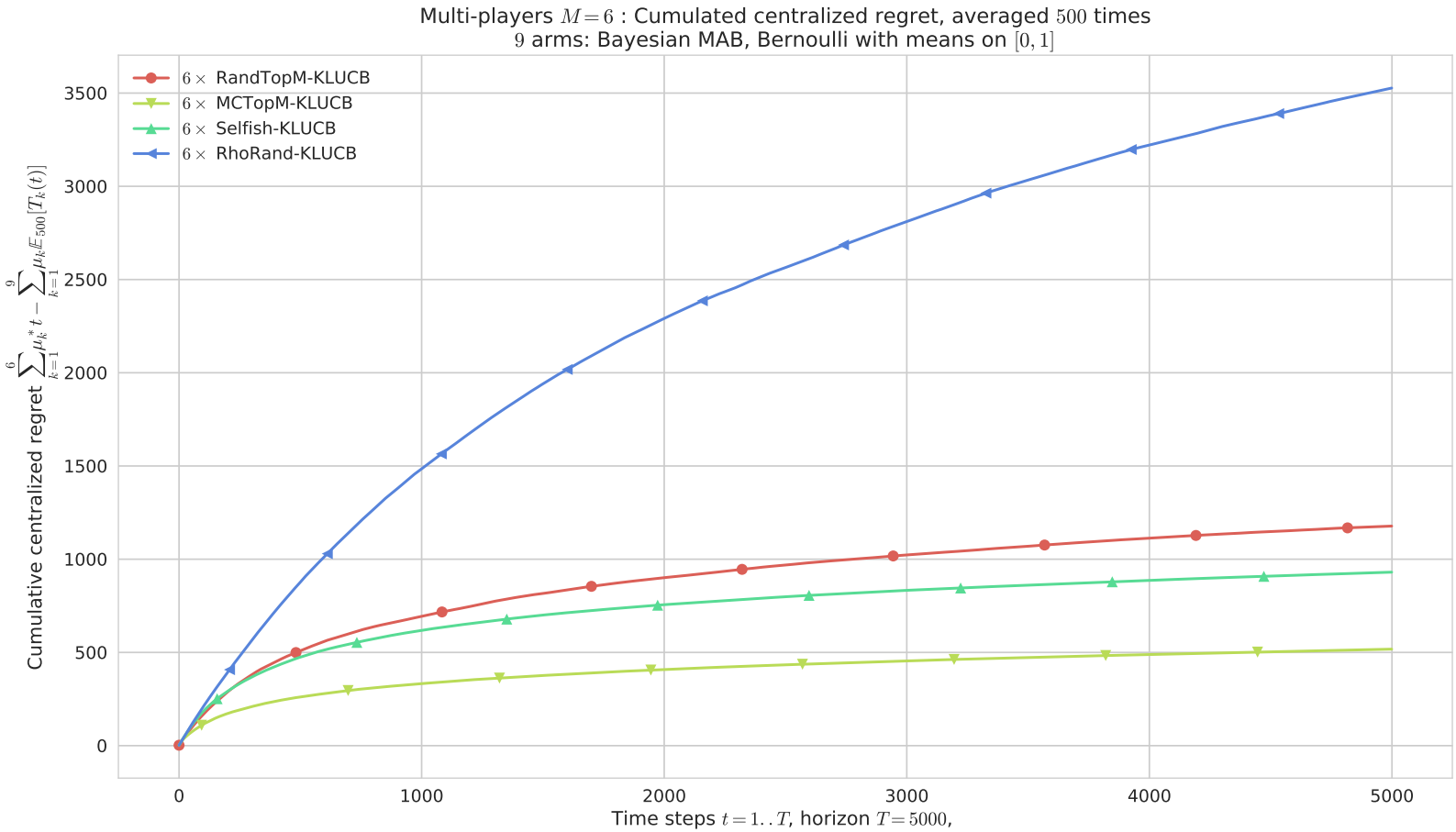


Figure 6: Regret, $M = 6$ players, $K = 9$ arms, horizon $T = 5000$, against 500 problems $\boldsymbol{\mu}$ uniformly sampled in $[0, 1]^K$.
RhoRand < RandTopM < Selfish < MCTopM in most cases.

Regret upper-bound for MCTopM-kl-UCB

Theorem 2 [Besson & Kaufmann, 2017]

- If all M players use MCTopM-kl-UCB, then for any non-degenerated problem $\boldsymbol{\mu}$,

$$R_T(\boldsymbol{\mu}, M, \rho) \leq G_{M,\boldsymbol{\mu}} \log(T) + o(\log T) \, .$$

Remarks

- Hard to prove, we had to carefully design the MCTopM algorithm to conclude the proof,
- For the suboptimal selections, we *match our lower-bound* !
- We also *minimize the number of channel switching*: interesting as it costs energy,
- Not yet possible to know what is the best possible control of collisions...

In this model

The Selfish decentralized approach = device don't use sensing, just learn on the receive acknowledgement,

- Like our first IoT model,
- It works fine in practice!
- Except... when it fails drastically!
- In small problems with M and $K = 2$ or 3, we found small probability of failures (*i.e.*, linear regret), and this prevents from having a generic upper-bound on regret for Selfish. Sadly...

Illustration of failing cases for Selfish

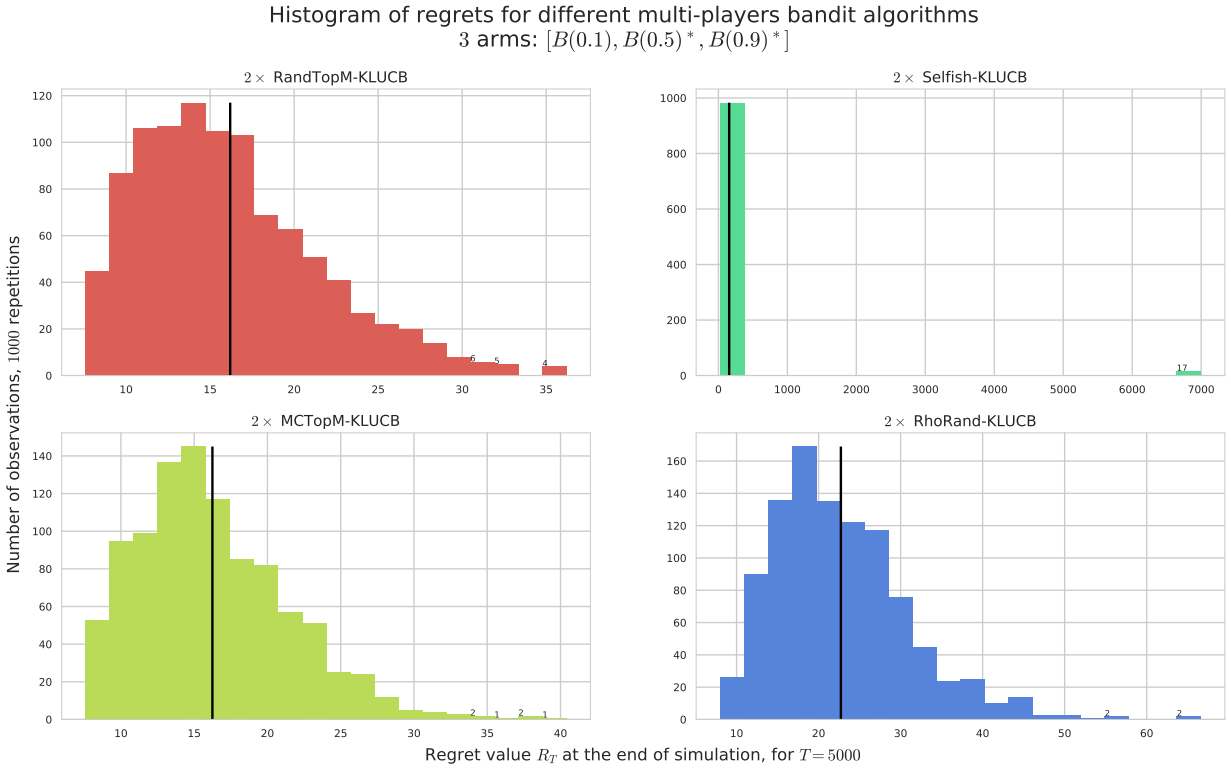


Figure 7: Regret for $M = 2$ players, $K = 3$ arms, horizon $T = 5000$, 1000 repetitions and $\boldsymbol{\mu} = [0.1, 0.5, 0.9]$. Axis x is for regret (different scale for each), and Selfish have a small probability of failure (17 cases of $R_T \geq T$, out of 1000). The regret for the three other algorithms is very small for this "easy" problem.

Perspectives

Theoretical results

- MAB algorithms have guarantees for *i.i.d. settings*,
- But here the collisions cancel the *i.i.d.* hypothesis,
- Not easy to obtain guarantees in this mixed setting (*i.i.d.* emissions process, "game theoretic" collisions).
- For OSA devices (always emitting), we obtained strong theoretical results,
- But harder for IoT devices with low duty-cycle...

Real-world experimental validation ?

- Radio experiments will help to validate this. Hard !

Other directions of future work

- *More realistic emission model*: maybe driven by number of packets in a whole day, instead of emission probability.
- Validate this on a *larger experimental scale*.
- Extend the theoretical analysis to the large-scale IoT model, first with sensing (*e.g.*, models ZigBee networks), then without sensing (*e.g.*, LoRaWAN networks).
- And also conclude the Multi-Player OSA analysis (remove hypothesis that objects know M, allow arrival/departure of objects, non-stationarity of background traffic etc)

Conclusion I

We showed

- Simple Multi-Armed Bandit algorithms, used in a Selfish approach by IoT devices in a crowded network, help to quickly learn the best possible repartition of dynamic devices in a fully decentralized and automatic way,
- For devices with sensing, smarter algorithms can be designed, and analyze carefully.
- Empirically, even if the collisions break the *i.i.d* hypothesis, stationary MAB algorithms (UCB, TS, kl-UCB) outperform more generic algorithms (adversarial, like Exp3).

Conclusion II

But more work is still needed...

- **Theoretical guarantees** are still missing for the IoT model, and can be improved (slightly) for the OSA model.
- Maybe study **other emission models**.
- Implement this on **real-world radio devices** (*TestBed*).

Thanks!

Any question?